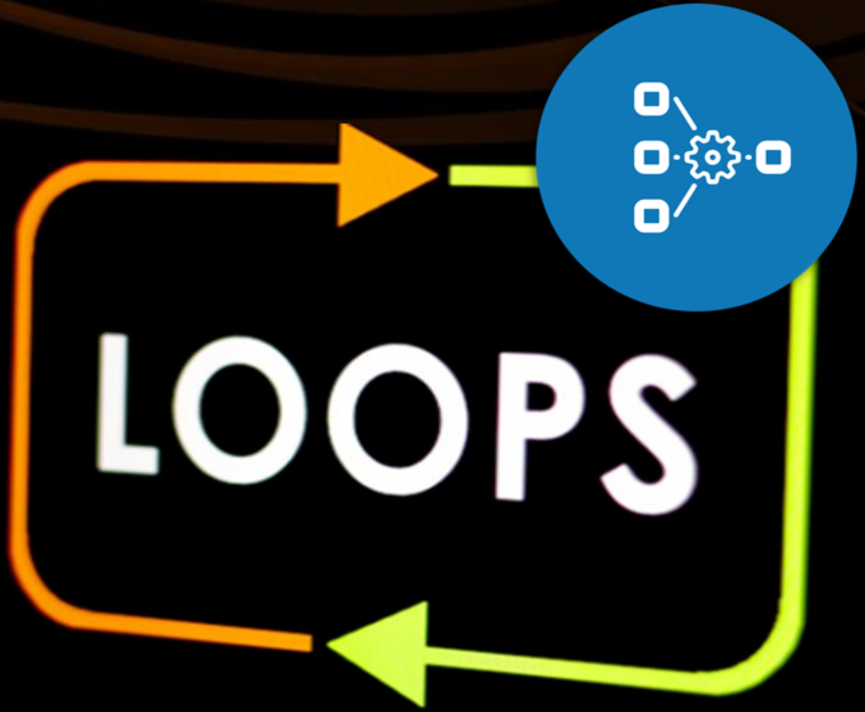
Работа с по-сложни цикли

Цикли със стъпка, While, Do…While





СофтУни

трейнърски екип

Софтуерен университет

http://softuni.bg

Have a Question?

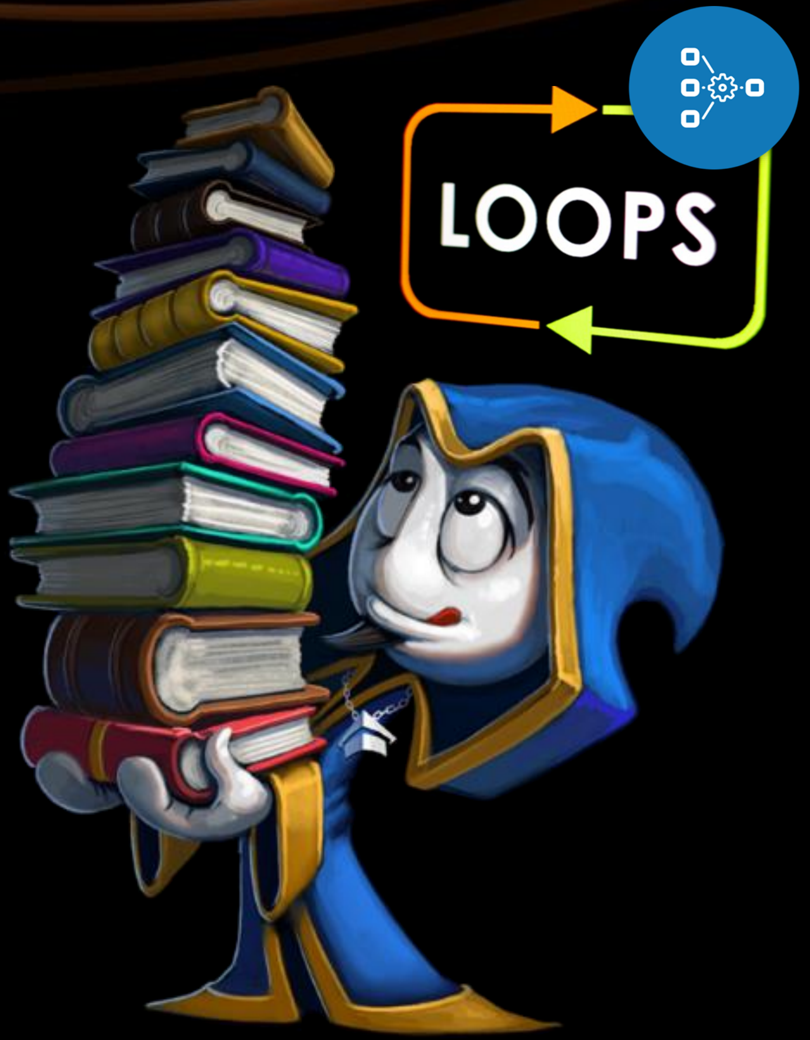
SOFTWARE UNIVERSITY FOUNDATION

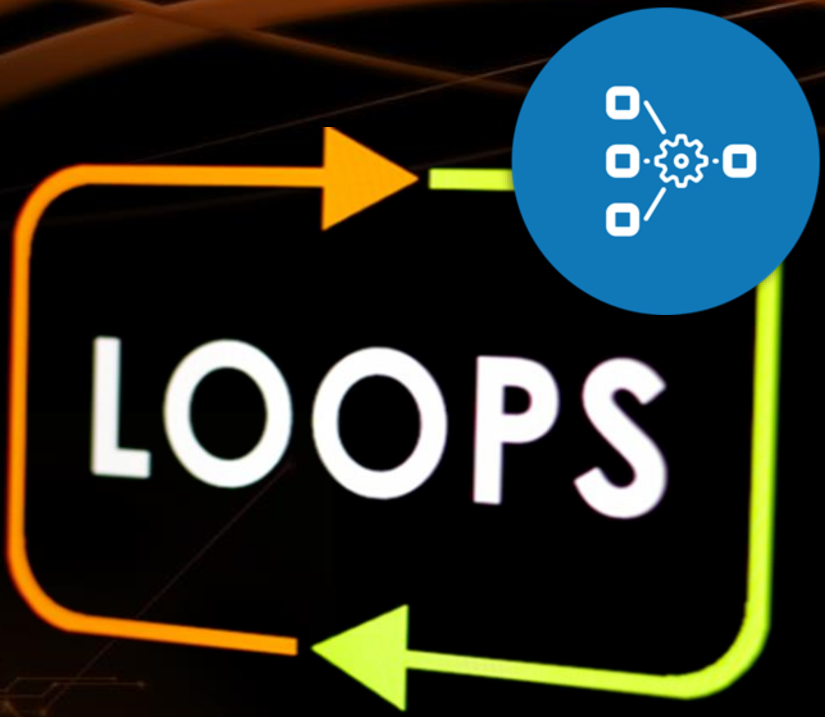
sli.do

12344321

Съдържание

- По-сложни конструкции за цикъл:
  - Цикъл със стъпка
  - Цикъл с намаляваща стъпка
  - Цикъл - **while**
  - Цикъл - **do-while**
  - Безкраен цикъл
    - Оператор **break**
    - Оператор **continue**









Цикли със стъпка

Работа с по-сложни **for**-цикли

Числата от 1 до N през 3 - условие

SOFTWARE UNIVERSITY FOUNDATION

- Напишете програма, която:
  - Прочита цяло число **n**
  - Отпечатва числата от **1** до **n** със стъпка **3**
- Примерен вход и изход:

Вход	Изход
10	1, 4, 7, 10
15	1, 4, 7, 10, 13



Числата от 1 до N през 3 – решение

```
var n = int.Parse(Console.ReadLine());
for (var i = 1; i <= n; i+=3)
{
    Console.WriteLine(i);
}
```

Задаване на стъпка

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/156#0

Числата от N до 1 в обратен ред - условие

- Напишете програма, която:
  - Прочита цяло число **n**
  - Отпечатва числата от **n** до **1** в обратен ред (стъпка -1)
- Примерен вход и изход:

Вход	Изход
`100`	`100, 99, 98, …, 3, 2, 1`
`3`	`3, 2, 1`

Числата от N до 1 в обратен ред – решение

```
var n = int.Parse(Console.ReadLine());
for (var i = n; i >= 1; i-=1)
{
  Console.WriteLine(i);
}
```

Отрицателна стъпка: `-1`

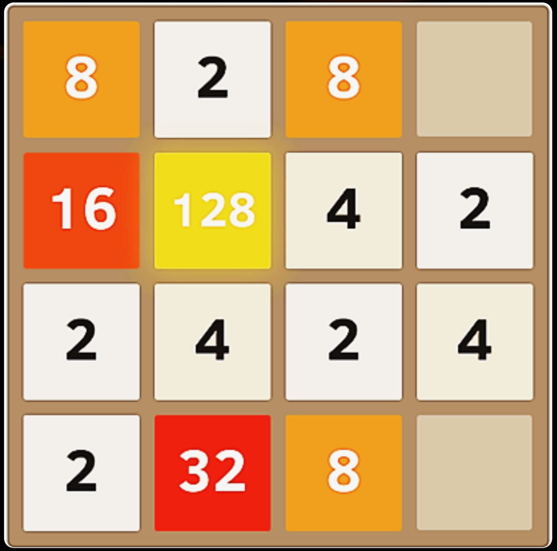
Обърнато условие: `i >= 1`



Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/156#1

Числата от 1 до 2^n с for-цикъл – условие

- Напишете програма, която:
  - Прочита цяло число **n**
  - Отпечатва числата от 1 до 2^n
- Примерен вход и изход:

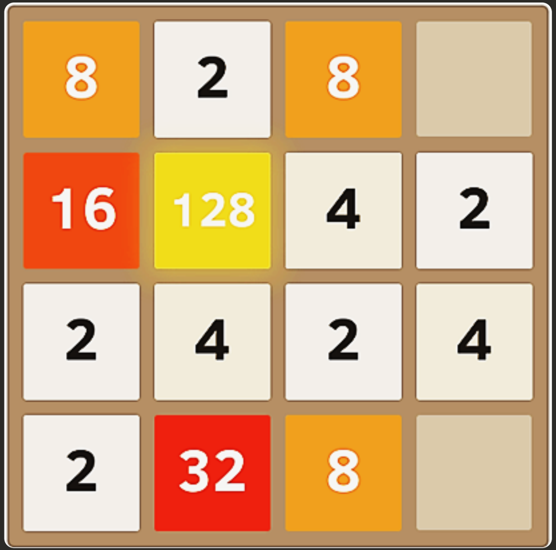


Вход		Изход
`10`	→	`1, 2, 4, 8, 16, 32, …, 1024`
`5`	→	`1, 2, 4, 8, 16, 32`

Числата от 1 до 2^n с for-цикъл – решение

SOFTWARE UNIVERSITY FOUNDATION

```
var n = int.Parse(Console.ReadLine());
var num = 1;
for (var i = 0; i <= n; i++)
{
    Console.WriteLine(num);
    num = num * 2;
}
```



Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/156#2

Четни степени на 2 - условие

- Напишете програма, която:
  - Прочита цяло число **n**
  - Отпечатва четните степени на **2** до **2^n**: 2^0, 2^2, 2^4, 2^8, …, **2^n**
- Примерен вход и изход:

Вход	Изход
`10`	`1, 4, 16, …, 1024`
`3`	`1, 4`

Четни степени на 2 – решене

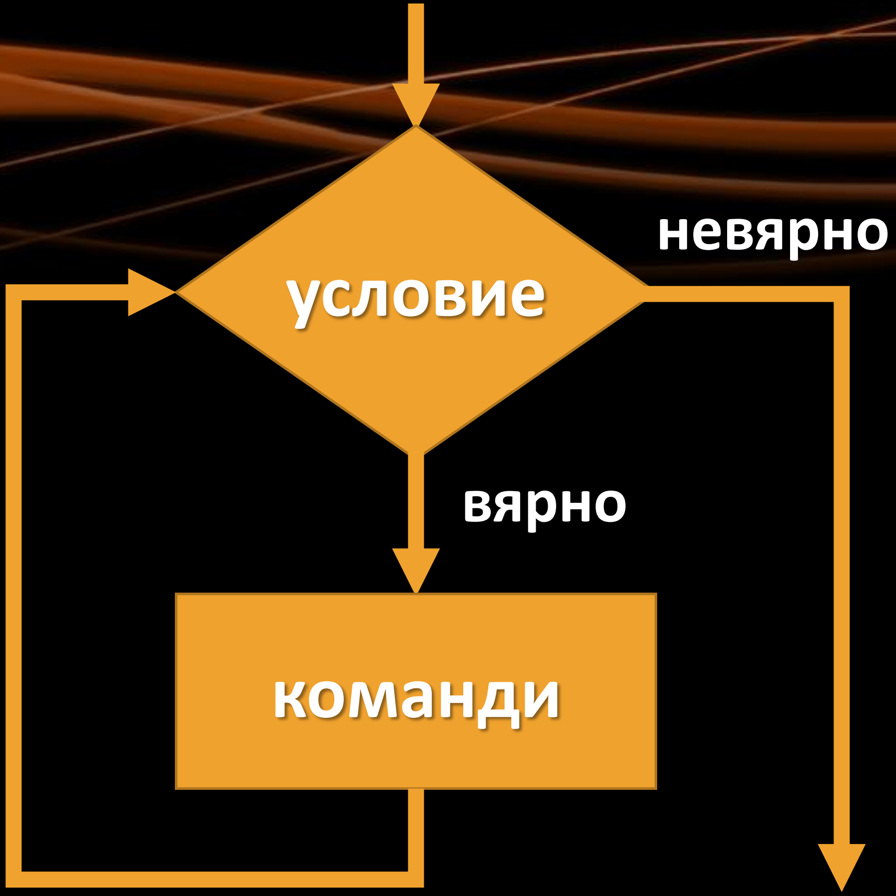
SOFTWARE UNIVERSITY FOUNDATION

```
var n = int.Parse(Console.ReadLine());
var num = 1;
for (var i = 0; i <= n; i+=2)
{
    Console.WriteLine(num);
    num = num * 2 * 2;
}
```

Ползваме стъпка 2

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/156#3







While цикъл

Повторение докато е в сила дадено условие

While цикъл

- Тялото на цикъла се изпълнява докато е вярно дадено условие

```
while (...)
{
    //code
}
```

Условие (true/false)

Код за изпълнение (повторение)

Редица числа 2k+1 - условие

- Напишете програма, която:
  - Прочита цяло число **n**
  - Отпечатва всички числа ≤ **n** от редицата: 1, 3, 7, 15, 31, …
  - Всяко следващо число е равно на предишното * 2 + 1

```
1, (1*2)+1 = 3, (3*2)+1 = 7, (7*2)+1 = 15 …
```

Редица числа 2k+1 – решение

```
var n = int.Parse(Console.ReadLine());
var num = 1;
while (num <= n)
{
    Console.WriteLine(num);
    num = 2 * num + 1;
}
```

Повторение докато е в сила условието num ≤ n

1, 3, 7, 15, 31, 63, …

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/156#4

Число в диапазона [1…100] - условие

- Напишете програма, която:
  - Прочита цяло число
  - Проверява дали е в диапазона [1…100]
  - При:
    - Намиране на число в диапазона, прекратява изпълнение
    - Невалидно число прочита ново



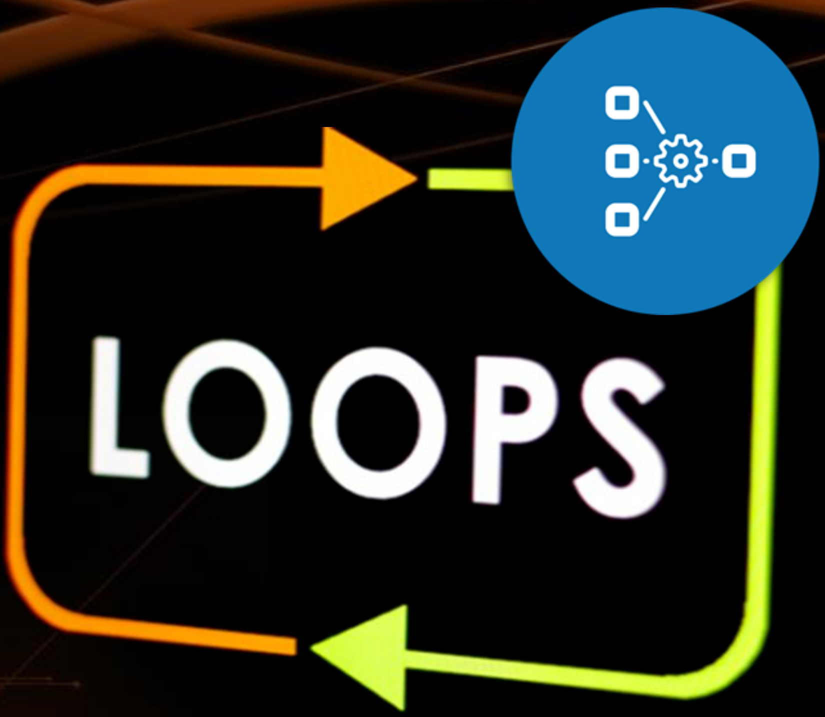
Число в диапазона [1…100] – решение

```
var num = int.Parse(Console.ReadLine());
while (num < 1 || num > 100)
{
   Console.WriteLine("Invalid number!");
   num = int.Parse(Console.ReadLine());
}

Console.WriteLine("The number is: {0}", num);
```



Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/156#5

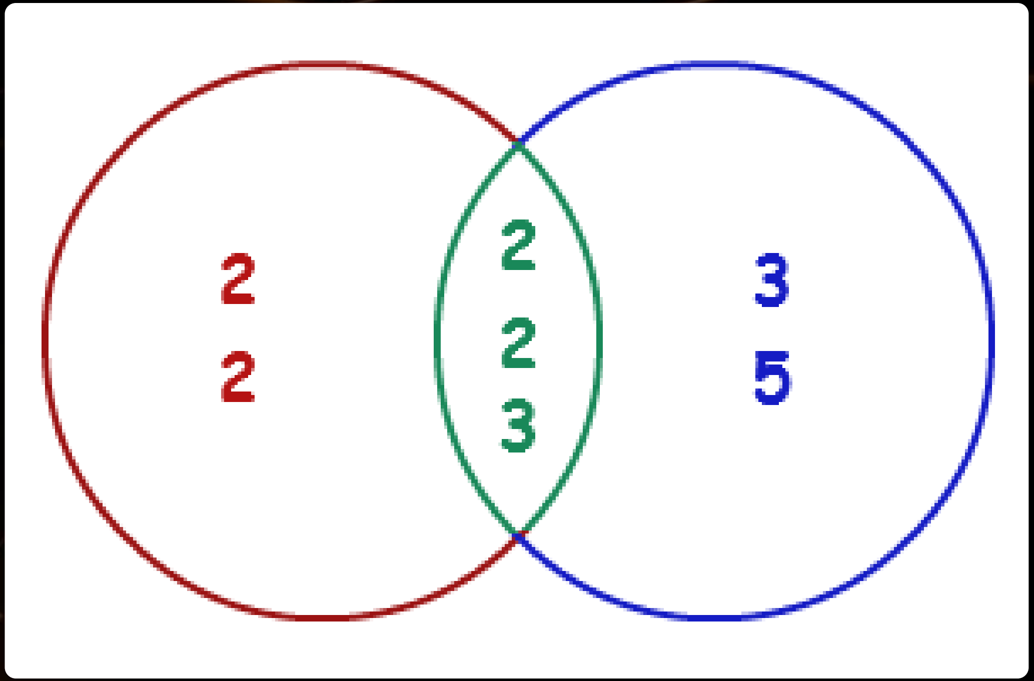






Цикли със стъпка и while цикъл

Работа на живо в клас (лаб)





Най-голям общ делител (НОД)

Алгоритъм на Евклид

Най-голям общ делител (НОД)

- Най-голям общ делител (НОД) на две естествени числа **a** и **b** е най-голямото число, което дели едновременно **a** и **b** без остатък
  - НОД(24, 16) = 8
  - НОД(67, 18) = 1
  - НОД(12, 24) = 12
  - НОД(15, 9) = 3
  - НОД(10, 10) = 10
  - НОД(100, 88) = 4



Алгоритъм на Евклид за НОД - условие

- Напишете програма, която:
  - Прочита 2 цели числа **a** и **b**
  - Намира най-големия им общ делител - **НОД**(**a**, **b**)
- Насоки:
  - Докато не се достигне остатък 0:
    - Дели се по-голямото число на по-малкото
    - Взема се остатъка от делението

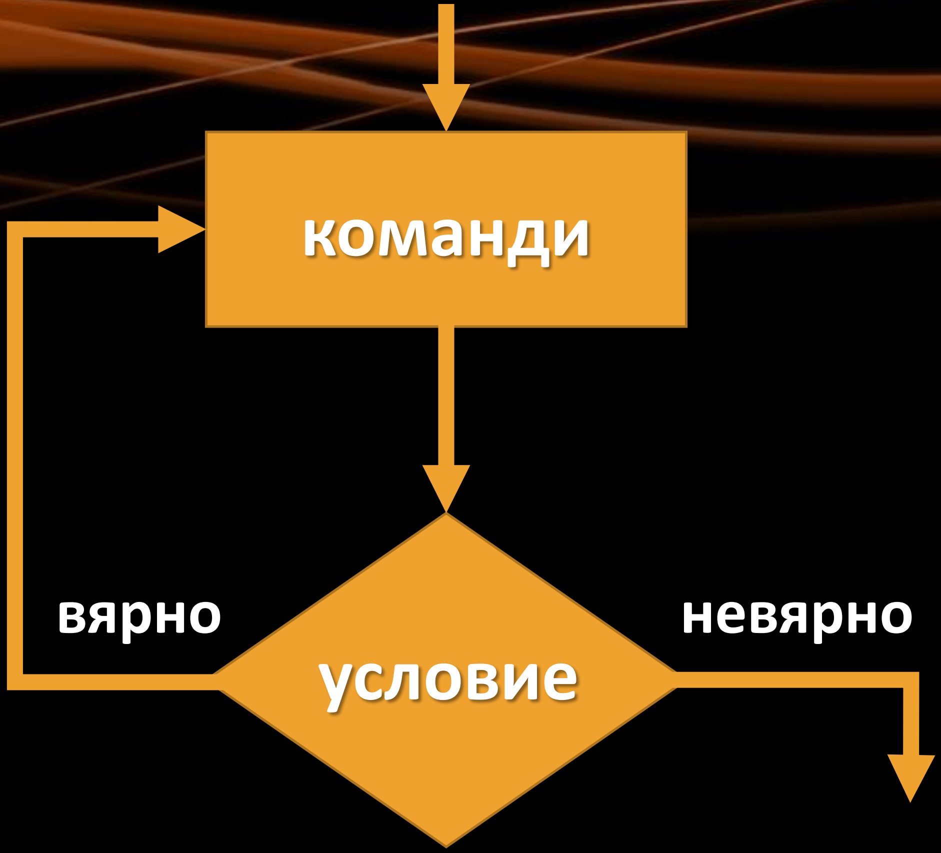
```
while b ≠ 0
  var oldB = b;
  b = a % b;
  a = oldB;
print a;
```

Алгоритъм на Евклид за НОД


```
var a = int.Parse(Console.ReadLine());
var b = int.Parse(Console.ReadLine());
while (b != 0)
{
    var oldB = b;
    b = a % b;
    a = oldB;
}
Console.WriteLine("GCD = {0}", a);
```

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/156#6







Do…While цикъл

Повторение докато е изпълнено условието

Do-while цикъл

- Тялото на цикъла се изпълнява докато е вярно дадено условие
  - Изпълнява се минимум един път

```
do
{
   //code
} while (...);
```

Код за изпълнение (повторение)

Условие (true/false)

Изчисляване на факториел - условие

- Напишете програма, която:
  - Прочита естествено число **n**
  - Изчислява факториел от n (n!)
- Примерен вход и изход:
  - 5! = 1 * 2 * 3 * 4 * 5 = 120

Вход		Изход
5	→	120

n!

Изчисляване на факториел – решение

```
var n = int.Parse(Console.ReadLine());
var fact = 1;
do
{
    fact = fact * n;
    n--;
} while (n > 1);
Console.WriteLine(fact);
```

n!

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/156#7

Сумиране на цифрите на число - условие

- Напишете програма, която:
  - Прочита цяло положително число **n**
  - Сумира цифрите на **n**
- Примерен вход и изход:
  - **n** = 5634: 5 + 6 + 3 + 4 = 18

Вход	Изход
5634	18
1234	10

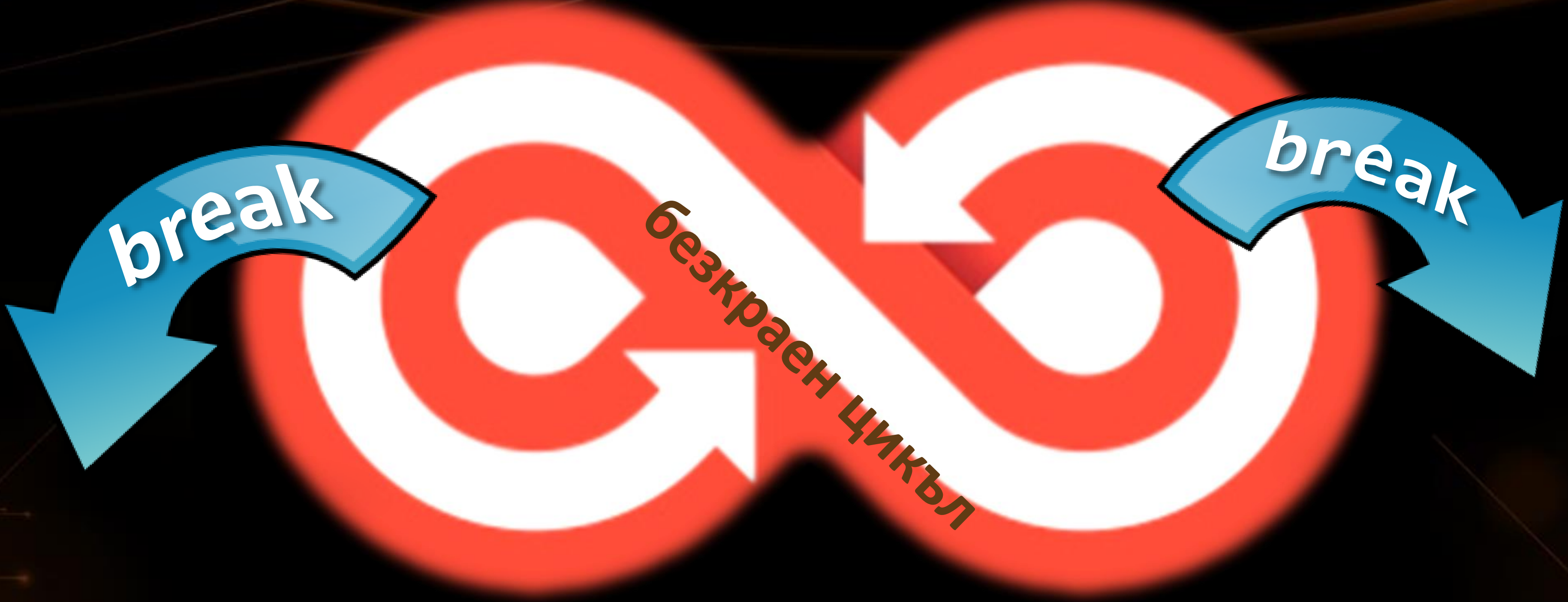
Сумиране на цифрите на число

```
var n = int.Parse(Console.ReadLine());
var sum = 0;
do
{
    sum = sum + (n % 10);
    n = n / 10;
} while (n > 0);
Console.WriteLine("Sum of digits: {0}", sum);
```

n % 10 връща последната цифра на числото n

n / 10 премахва последната цифра на n

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/156#8



Безкрайни цикли и оператори `break` и `continue`

Безкраен цикъл

- Безкраен цикъл имаме когато:
  - Нямаме условие, което да прекрати цикъла
  - Нямаме команда, която да прекрати цикъла

```
while(true){
    Console.WriteLine("Infinite loop");
}
```

```
for (;;){
    Console.WriteLine("Infinite loop");
}
```

Условия за прекратяване на цикъл

```
while(...)
{
    Console.WriteLine("Infinite loop");
}
```

Условие за прекратяване на цикъл



```
for (;...;)
{
    Console.WriteLine("Infinite loop");
}
```

Условие за прекратяване на цикъл



Команда за прекратяване на цикъл

- Оператор break – прекъсване на цикъла

```
while(true)
{
    Console.WriteLine("Infinite loop");
    if (…)
    {
        break;
    }
}
```

Условие за прекратяване на цикъл

Команда за излизане от цикъл

Прости числа - условие

- Напишете програма, която:
  - Прочита цяло число **n**
  - Проверява да ли **n** е просто число
- Насоки:
  - Едно число **n** е просто, ако се дели единствено на **1** и **n** и е по-голямо от 1
  - Прости числа: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, …
  - Непрости (композитни) числа: 10 = 2 * 5, 21 = 3 * 7, 143 = 13 * 11

Прости числа – решение

```
var n = int.Parse(Console.ReadLine());
var prime = true;
for (var i = 2; i <= Math.Sqrt(n); i++)
    if (n % i == 0) {
        prime = false;
        break;
    }
if (prime) Console.WriteLine("Prime");
else Console.WriteLine("Not prime");
```

break излиза от цикъла

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/156#9

Четно число - условие

- Напишете програма, която:
  - Прочита число **n**
  - Проверява дали **n** е четно
  - При невалидно число се връща към повторно въвеждане

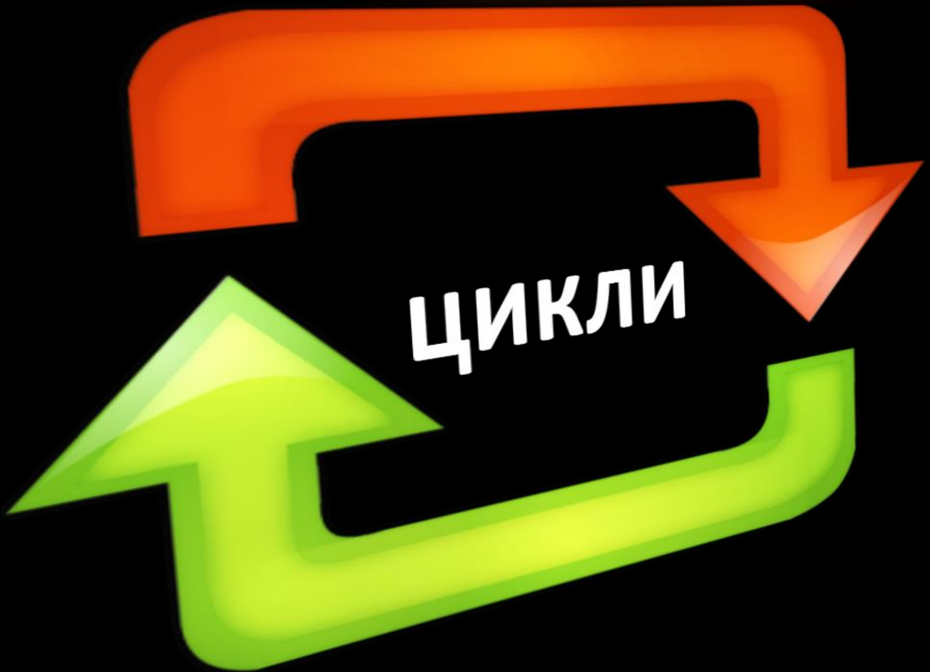


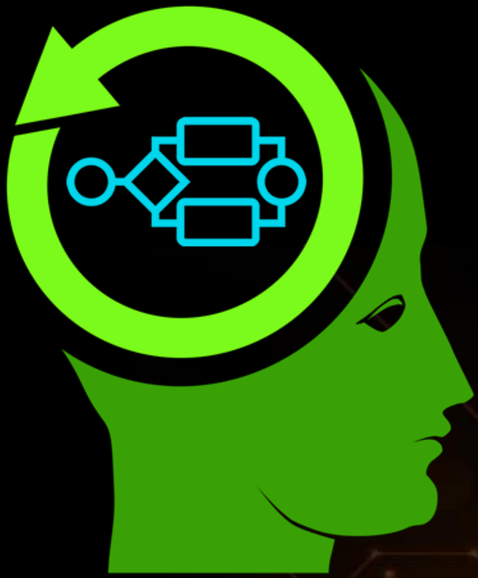
Четно число – решение

```
while (true)
{
   Console.Write("Enter even number: ");
   var n = int.Parse(Console.ReadLine());
   if (n % 2 == 0)
      break; // even number -> exit from the loop
   Console.WriteLine("The number is not even.");
}

Console.WriteLine("Even number entered: {0}", n);
```







Задачи с цикли

Числа на Фибоначи - условие

- Напишете програма, която:
  - Прочита число **n**
  - Пресмята **n-тото** число на Фибоначи
- Числата на Фибоначи са следните: 1, 1, 2, 3, 5, 8, 13, 21, 34, …
  - $F_0 = 1$
  - $F_1 = 1$
  - $F_n = F_{n-1} + F_{n-2}$
- Примерен вход и изход:
  - F(15) = 987

Вход	Изход
15	987
3	2

Числа на Фибоначи

```
var n = int.Parse(Console.ReadLine());
var f0 = 1;
var f1 = 1;
for (var i = 0; i < n-1; i++)
{
   var fNext = f0 + f1;
   f0 = f1;
   f1 = fNext;
}
Console.WriteLine(f1);
```



Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/156#11

Пирамида от числа – условие

- Да се отпечатат **числата 1…n в пирамида** като в примерите:

```
n = 7
```

```
1
2 3
4 5 6
7
```

```
n = 10
```

```
1
2 3
4 5 6
7 8 9 10
```

```
n = 12
```

```
1
2 3
4 5 6
7 8 9 10
11 12
```

```
n = 15
```

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```



Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/156#12

Пирамида от числа – решение

```
var n = int.Parse(Console.ReadLine());
var num = 1;
for (var row = 1; row <= n; row++)
{
    for (var col = 1; col <= row; col++)
    {
        if (col > 1) Console.Write(" ");
        Console.Write(num);
        num++;
        if (num > n) break;
    }
    Console.WriteLine();
    if (num > n) break;
}
```



Таблица с числа – условие

- Да се отпечатат **числата 1…n в таблица** като в примерите:

```
n = 2
```

```
1 2
2 1
```

```
n = 3
```

```
1 2 3
2 3 2
3 2 1
```

```
n = 4
```

```
1 2 3 4
2 3 4 3
3 4 3 2
4 3 2 1
```

```
n = 5
```

```
1 2 3 4 5
2 3 4 5 4
3 4 5 4 3
4 5 4 3 2
5 4 3 2 1
```



Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/156#13

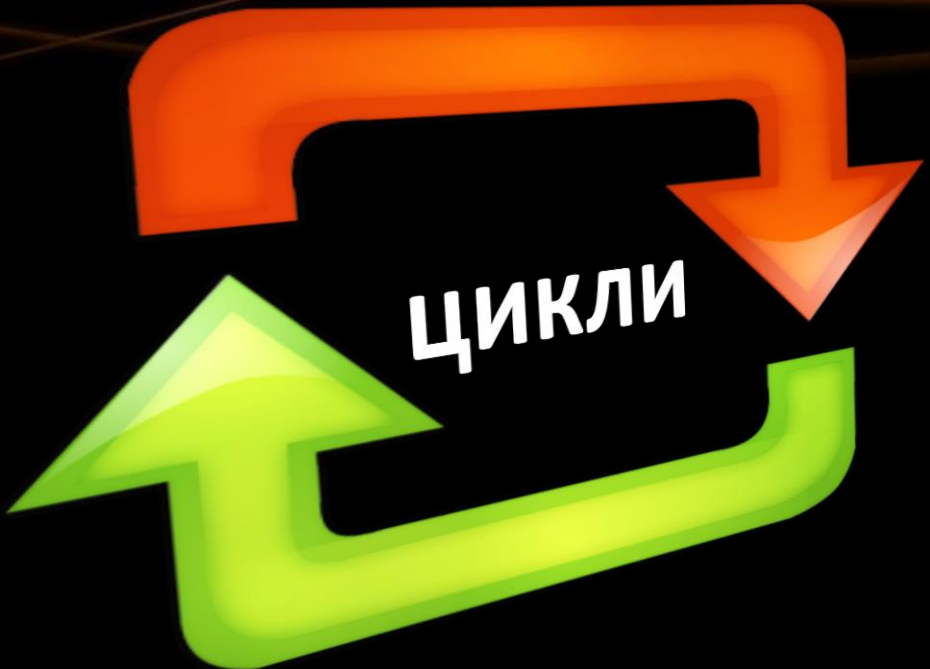
Таблица с числа – решение

```
var n = int.Parse(Console.ReadLine());
for (var row = 0; row < n; row++)
{
    for (var col = 0; col < n; col++)
    {
        var num = row + col + 1;
        if (num > n) num = 2 * n - num;
        Console.Write(num + " ");
    }
    Console.WriteLine();
}
```

```
1 2 3 4 5
2 3 4 5 4
3 4 5 4 3
4 5 4 3 2
5 4 3 2 1
```

Тестване на решението: https://judge.softuni.bg/Contests/Practice/Index/156#13







По-сложни задачи с цикли

Работа на живо в клас (лаб)

Какво научихме днес?

- Можем да ползваме **for**-цикли със **стъпка**:

```
for (var i = 1; i <= n; i+=3)
    Console.WriteLine(i);
```



- Цикли **while** / **do-while** повтарят блок от код докато е в сила дадено условие:

```
var num = 1;
while (num <= n)
    Console.WriteLine(num++);
```

Какво научихме днес? (2)

- Можем да създаваме **безкрайни** цикли и когато се наложи да излизаме от тях:

```
for (;;) {
    if (…)
        break;
}
```

```
while (true){
    if (…)
        break;
}
```



Чертане с цикли



Въпроси?

https://softuni.bg/courses/programming-basics/

Лиценз

- Настоящият курс (слайдове, примери, видео, задачи и др.) се разпространяват под свободен лиценз "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International"

- Благодарности: настоящият материал може да съдържа части от следните източници
  - Книга "Основи на програмирането със C#" от Светлин Наков и колектив с лиценз CC-BY-SA

Безплатни обучения в СофтУни

- Фондация "Софтуерен университет" – softuni.org
- Софтуерен университет – качествено образование, професия и работа за софтуерни инженери
  - softuni.bg
- СофтУни @ Facebook
  - facebook.com/SoftwareUniversity
- СофтУни форуми – forum.softuni.bg

